REMARKS/ARGUMENTS

Claims 1-24 are pending in the present application. Claims 1-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Morrison et al. (Pub. No.: US 2002/0038398) in view of Nakamura (US Patent 5,850,529).

Applicant's respectfully submit that the cited references do not teach, suggest or disclose "[a] method for executing a locked bus transaction in a multi-node system, comprising: initiating a locked-bus transaction at a bus agent; transmitting a locked-bus request to a first node controller; and deferring the locked-bus transaction at the bus agent by said first node controller" (e.g. as recited in claim 1).

The Office Action asserts:

> Morrison disclose all the limitations as above except deferring the locked-bus transaction at the bus agent by said first node controller. However, Nakamura discloses controller detects whether the PCI bus is in a state of resource lock by using the lock signal at the granted bus access enable signal and prohibiting controller from executing a transaction when PCI bus is in a state or resource lock. Therefore, the target retry generated. (col. 2, lines 38-65). Furthermore, Nakamura discloses a target retry is sent from a target device. (emphasizing at the bus agent as applicant claimed, col. 13, lines 1-7).

Column 2 lines 38-65 state –

> There has also been provided, in accordance with yet another aspect of the present invention, a method of detecting a state of a resource lock by DMA controller means on a PCI bus, on which a lock signal line indicating whether an access to a resource is exclusively executed and a bus access request signal line and a bus access enable signal line are allocated into each of PCI agent devices, the method of comprising the steps of transmitting a bus access request signal from an I/O expansion device through a bridge device to bus arbiter means, receiving a granted bus access enable signal to the DMA controller means by the bus arbiter means, determining whether the PCI bus is in a state of resource lock by using the lock signal at the granted bus access enable signal, and prohibiting the DMA controller means from executing a transaction for DMA transfer, when the PCI bus is in a state of resource lock.
>
> In the computer system of this invention, the DMA controller detects whether there is a resource put into the state of lock by other bus masters on the PCI bus. If it is not in a state of resource locking, the DMA controller does not perform the transaction for beginning DMA transfer. Therefore, the target retry generated in the transaction for the DMA transfer is caused other than the bus lock. Thus, even if the DMA controller does not release the bus access by the target retry, as a result, a memory device is not permanently locked. Therefore, it can prevent the DMA transfer from malfunctioning in view of the generation of the target retry.

Column 13 lines 1-7 state:

> The resource lock detection logic device controls the state transition explained as shown in FIG. 4 by using LOCK# and FRAME#. If internal PCI bus 2 is in a state of the resource lock, the resource lock detection logic device informs ISA DMAC 1521 of the resource lock state, preventing ISA DMAC 1521 from executing the transaction for DMA transfer. Specifically, the resource lock detection logic device prevents DACK# and FRAME# output from ISA DMAC 1521, and makes REQ# inactive and invalidates GNT#.

Applicants respectfully submit that the cited references do not teach suggest or disclose "[a] method for executing a locked bus transaction in a multi-node system, comprising ... *deferring* the locked-bus transaction at the bus agent by said first node controller" (emphasis added). Specifically, neither Morrison nor Nakamura teach, suggest or disclose anywhere the deferring of the locked bus transaction (deferring meaning delaying *and* retrying until the locked bus transaction is accepted). Further support of the operation of deferring can be found at page 7 line 16 which states: "In block 203, the node controller 115 does not accept this locked-bus transaction *(e.g., by asserting an appropriate signal, DEFER# in this embodiment, and by giving a retry response), causing the processor to retry the locked-bus transaction until it is accepted*" (emphasis added).

The Office Action states "...Nakamura discloses controller detects whether the PCI bus is in a state of resource lock by using the lock signal at the granted bus access enable signal and prohibiting controller from executing a transaction when PCI bus is in a state or resource lock. Therefore, the target retry generated. (col. 2, lines 38-65)". Applicants respectfully dissent. Prohibiting the controller from executing a transaction does not correspond to a target retry. Unlike the various claimed embodiments (see also portions of specification cited above), neither Morrison nor Nakamura specifically disclose a retry response after instituting a locked-bus transaction at the bus agent by a

node controller. Additionally, Applicants respectfully submit that there is no suggestion or motivation to combine Morrison and Nakamura beyond the impermissible use of hindsight.

Since each and every limitation of the embodiment of claim 1 is not disclosed in, nor taught by cited references, reconsideration and withdrawal of the rejection of claim 1 under 35 U.S.C. § 103(a) is respectfully requested. Independent claims 10, 19, and 21 contain similar allowable limitations and therefore the rejection of these claims under 35 U.S.C. § 102(e) should be withdrawn as well. Dependent claims 2-9, 11-18, 20 and 22-24 depend from allowable independent claims, and therefore are in condition for allowance.

CONCLUSION

For at least the above reasons, Applicants respectfully submit that the present case is in condition for allowance and respectfully requests that the Examiner issue a notice of allowance.

The Office is hereby authorized to charge any fees determined to be necessary under 37 C.F.R. § 1.16 or § 1.17 or credit any overpayment to Kenyon & Kenyon **Deposit Account No. 11-0600**.

The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss any matter concerning this application.

Respectfully submitted,

Dated: August 16, 2004

Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

KENYON & KENYON
333 W. San Carlos Street
Suite 600
San Jose, CA 95110
Tel: (408) 975-7500
Fax: (408) 975-7501